Context Research

We’ll be using **[tflearn](http://tflearn.org/)**, a layer above **[tensorflow](https://www.tensorflow.org/)**, and of course [**Python**](https://www.python.org/). As always we’ll use **[iPython notebook](https://ipython.org/notebook.html)** as a tool to facilitate our work.

A chatbot framework needs a structure in which conversational intents are defined. One clean way to do this is with a JSON file, like [this](https://github.com/ugik/notebooks/blob/master/intents.json).

Each conversational intent contains:

* a **tag** (a unique name)
* **patterns** (sentence patterns for our neural network text classifier)
* **responses** (one will be used as a response)

And later on we’ll add some basic contextual elements.

For Adding Context we need to add below code,

ERROR\_THRESHOLD = 0.25

def classify(sentence):

# generate probabilities from the model

results = model.predict([bow(sentence, words)])[0]

# filter out predictions below a threshold

results = [[i,r] for i,r in enumerate(results) if r>ERROR\_THRESHOLD]

# sort by strength of probability

results.sort(key=lambda x: x[1], reverse=True)

return\_list = []

for r in results:

return\_list.append((classes[r[0]], r[1]))

# return tuple of intent and probability

return return\_list

def response(sentence, userID='123', show\_details=False):

results = classify(sentence)

# if we have a classification then find the matching intent tag

if results:

# loop as long as there are matches to process

while results:

for i in intents['intents']:

# find a tag matching the first result

if i['tag'] == results[0][0]:

# a random response from the intent

return print(random.choice(i['responses']))

# Structure of Intent File

If an intent wants to **set** context, it can do so:

*{“tag”: “rental”,  
“patterns”: [“Can we rent a moped?”, “I’d like to rent a moped”, … ],  
“responses”: [“Are you looking to rent today or later this week?”],****“context\_set”: “rentalday”*** *}*

If another intent wants to be contextually linked to a context, it can do that:

*{“tag”: “today”,  
“patterns”: [“today”],  
“responses”: [“For rentals today please call 1–800-MYMOPED”, …],****“context\_filter”: “rentalday”*** *}*

In this way, if a user just typed ‘today’ out of the blue (no context), our ‘today’ intent won’t be processed. If they enter ‘today’ as a response to our clarification question (intent tag:‘rental’) then the intent is processed.

*https://chatbotsmagazine.com/contextual-chat-bots-with-tensorflow-4391749d0077*